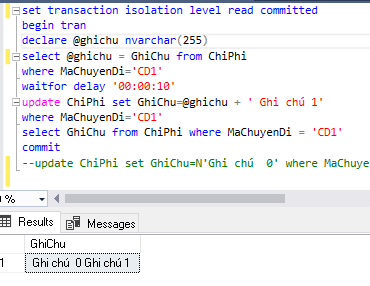
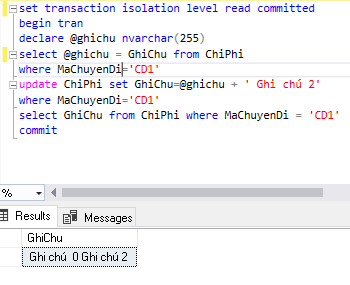
1. **Lost Data**

**Tình huống**

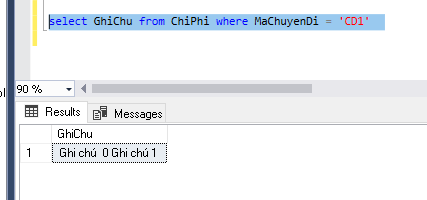
* Khi cả 2 giao tác T1, T2 cùng sửa trên 1 đơn vị dữ liệu Ghi chú trong bảng Chi phí (T1 chưa commit) khiến cho tác dụng của giao tác cập nhật của T1 sẽ đè lên tác dụng của giao tác cập nhật T2



(Ghi chú ban đầu là “Ghi chú 0” sau đó T1 update thêm thành “Không Ghi chú 1”)



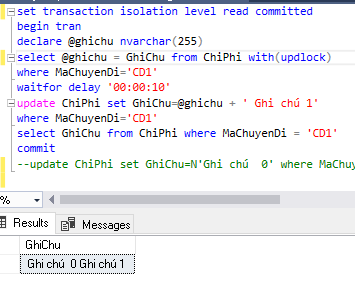
(Ghi chú ban đầu là “Không” sau đó T2 update thêm thành “Không Ghi chú 2”)

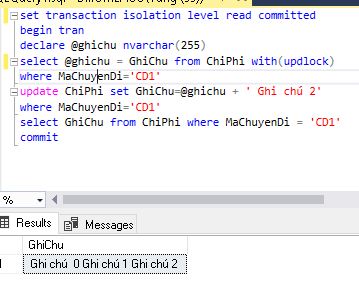


(Kết quả là T1 đè dữ liệu lên T2)

**Giải quyết**

* **Cách 1:** Khi sử dụng Read uncommitted hoặc Read committed, ta chuyển thành Update lock, từ đó T2 sẽ phải chờ T1 hoàn tất để nhả khóa

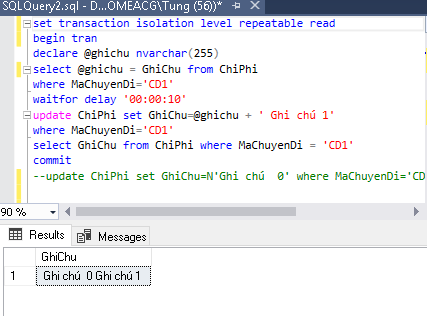


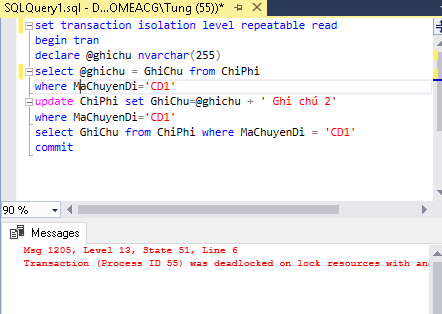


(Thêm Update Lock => giải quyết được )

* **Cách 2:** Sử dụng Repeatable Read => T2 chờ T1 thực hiện xong mới thực hiện update

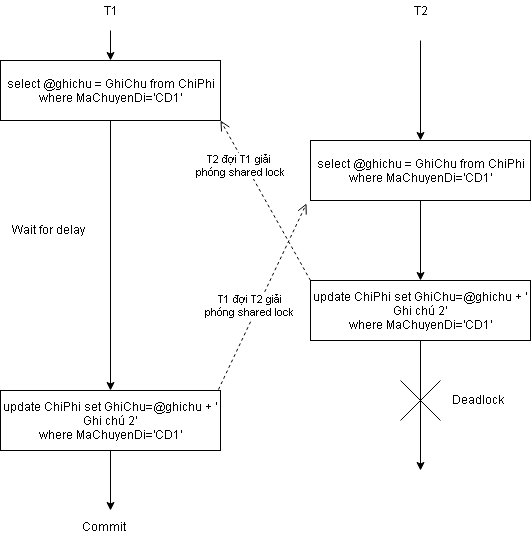
Vấn đề: Xảy ra deadlock



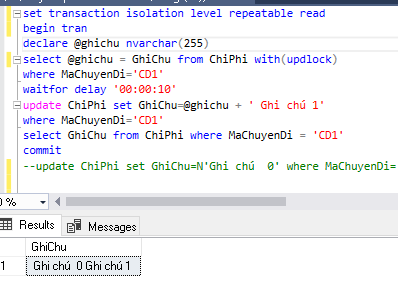


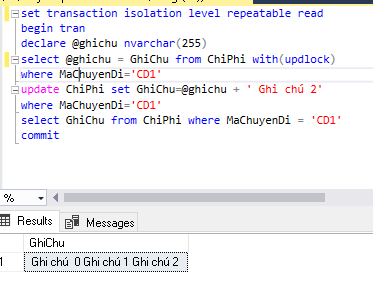
(Xảy ra deadlock)

**Nguyên nhân:**



**Giải quyết**: Thêm update lock vào lệnh select => T2 phải chờ T1 commit mới được thực hiện

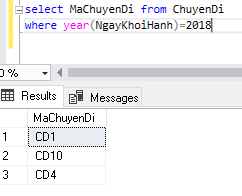




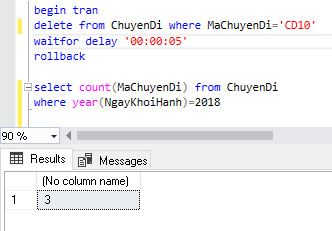
1. **Dirty Read**

**Tình huống**

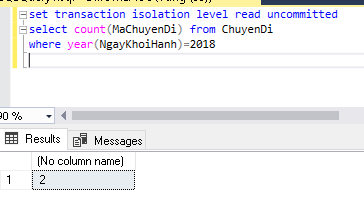
T1 đang xóa dữ liệu của chuyến đi CD10 thì T2 thực hiện đọc dữ liệu (đếm số chuyến đi trong năm 2018 có chuyến đi CD10). Lúc này T1 roll back nhưng T2 đã đếm sai số chuyến đi.



(Ban đầu có 3 chuyến đi)



(T1 roll back và đếm ra 3)

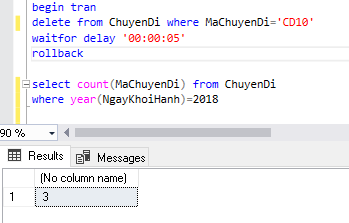


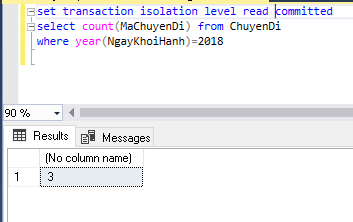
(T2 đếm sai ra 2)

**Nguyên nhân:** T2 đang là no lock nên k chờ T1 roll back mà đọc giá trị luôn

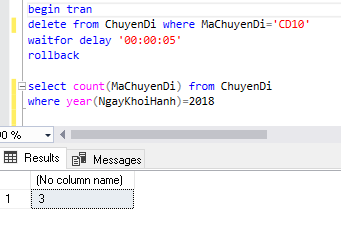
**Giải quyết:**

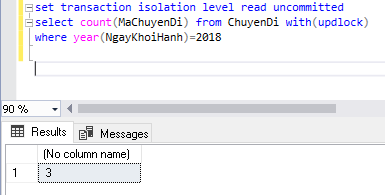
* **Cách 1:** Chuyển mức cô lập của T2 về Read committed, tức tạo khóa shared lock cho T2, shared lock sau khi đọc xong dữ liệu được giải phóng nên không cản trở đến thao tác cập nhật của T1 => T2 chờ T1 thực hiện xong giao tác rồi mới đọc





* **Cách 2:** Chuyển select của T2 từ No lock thành Update lock, T2 sẽ phải chờ T1 nhả khóa Exclusive lock => T2 phải chờ T1 hoàn tất mới thực hiện giao tác.

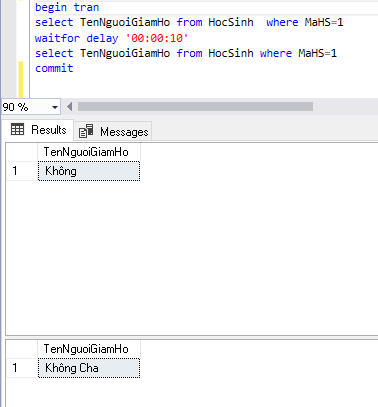


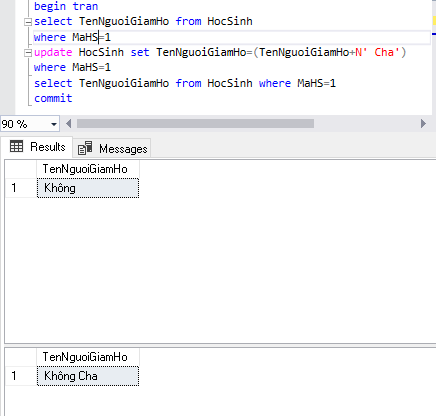


1. **Unrepeatable Data**

**Tình huống:**

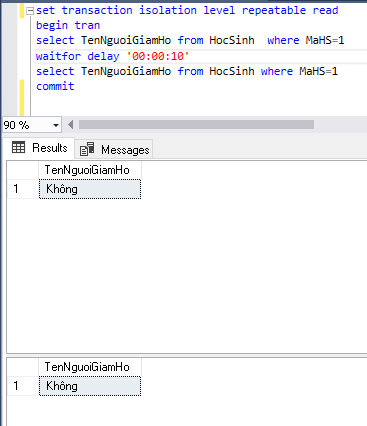
* Khi T1 đọc dữ liệu Tên người giám hộ của học sinh có mã học sinh là 1 thì tên người giám hộ là “Không” (nhưng chưa kịp commit) thì có giao tác T2 thực hiện cập nhật Tên người giám hộ thành “Không cha”
* Lúc này T1 đọc lại Tên người giám hộ không còn là “Không” nữa mà là “Không cha”

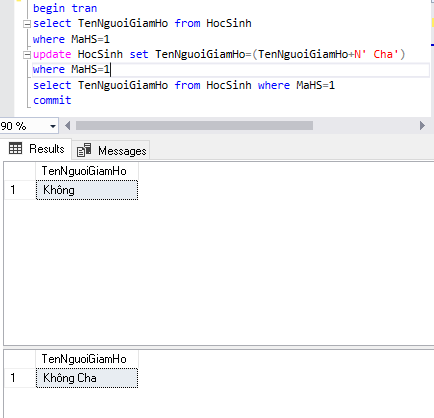




**Nguyên nhân:** Do T1 đang đọc mà T2 thực hiện update

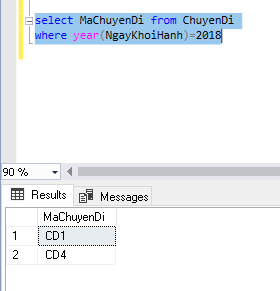
**Giải quyết:** sử dụng chế độ Repeatable Read, giữ shared lock đến hết giao tác => T1 thực hiện xong mới đến T2

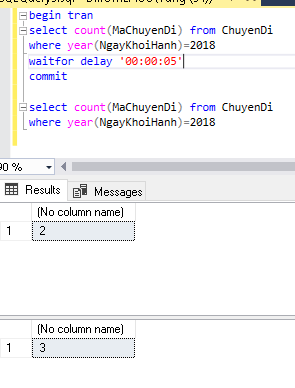


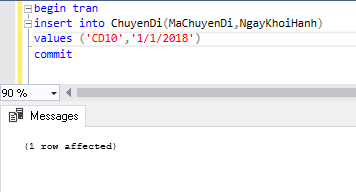


1. **Phantom**

**Tình huống:** T1 đang thực hiện đếm số chuyến đi trong năm 2018 thì T2 thực hiện insert thêm 1 chuyến đi trong năm 2018 làm T1 sai kết quả thao tác của T1



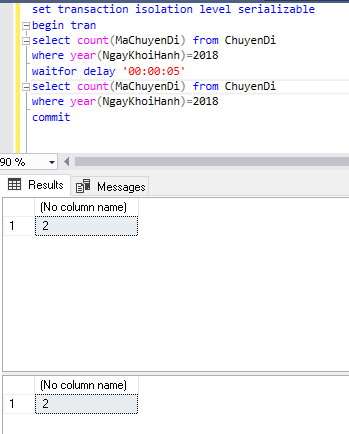


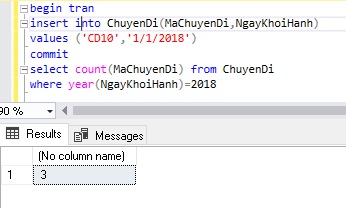


**Nguyên nhân:** T1 đang thực hiện đọc dữ liệu thì T2 cập nhật thêm dữ liệu làm việc đọc của T1 bị sai

**Giải quyết:** Chuyển mức cô lập của T1 về serializable. T1 không cho phép T2 Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock)

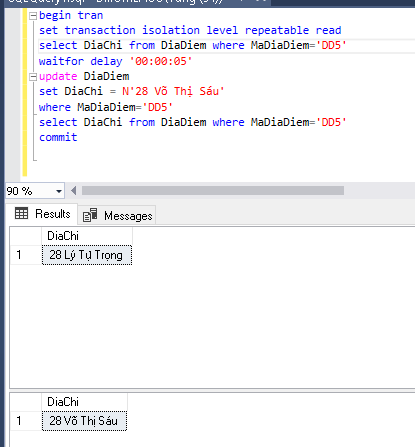
* T1 làm xong rồi mới tới T2 thao tác

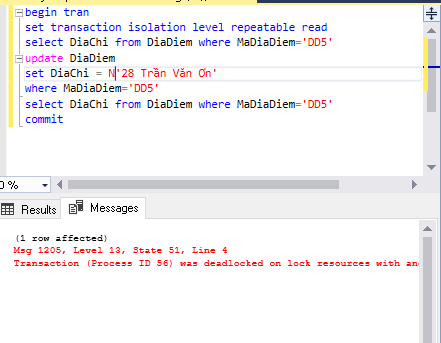




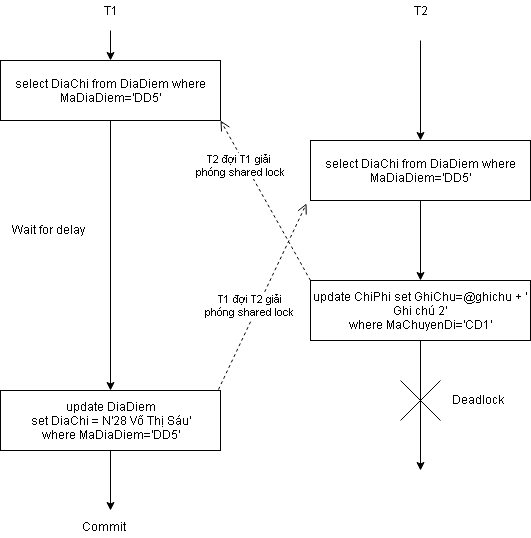
1. **Conversion Deadlock**

**Vấn đề:** T1 và T2 cùng xem địa điểm DD5 và thấy địa chỉ bị sai nên cả 2 đồng thời sửa. Mức cô lập là Repeatable read để chắc chắn giữ shared lock đến hết giao tác tránh xung đột. Nhưng lại xảy ra deadlock.





**Nguyên nhân:**



**Giải pháp:** Thêm Update lock vào lệnh select => T2 phải chờ T1 chạy xong mới được chạy.

